CS 475

Intro to Parallel Programming

Instructor: Matthew Meyn

Khuong Luu (luukh)

Project 7

1. **Computer I ran my program on**

I ran developed and ran my program in Computer Graphics Education Lab (CGEL)

Processor: Intel Xeon CPU E3-1230 v5 @ 3.40 GHz (8 CPUs)

Memory: 16384 MB

Operating System: Windows 10

GPU: NVIDIA GTX 1080 Ti

GPU memory: 19286 MB

1. **What dynamic thing did you do with the particle colors**

The color of a particle is determined by the sum of its current position and its current velocity, with respective to each components. Specifically, the formula is:

|  |
| --- |
| c.x = vp.x + pp.x;  c.y = vp.y + pp.y;  c.z = vp.z + pp.z;  c.w = 1.; |

When the particle bounce off the yellow sphere, its color is changed to briefly to **blue**.

|  |
| --- |
| if( IsInsideSphere( pp, Sphere1 ) ) {  vp = BounceSphere( p, v, Sphere1 );  pp = p + vp\*DT + (float4)(.5\*DT\*DT)\*G;  c.x = 0.0f;  c.y = 0.0f;  c.z = 0.9f;  c.w = 1.;  } |

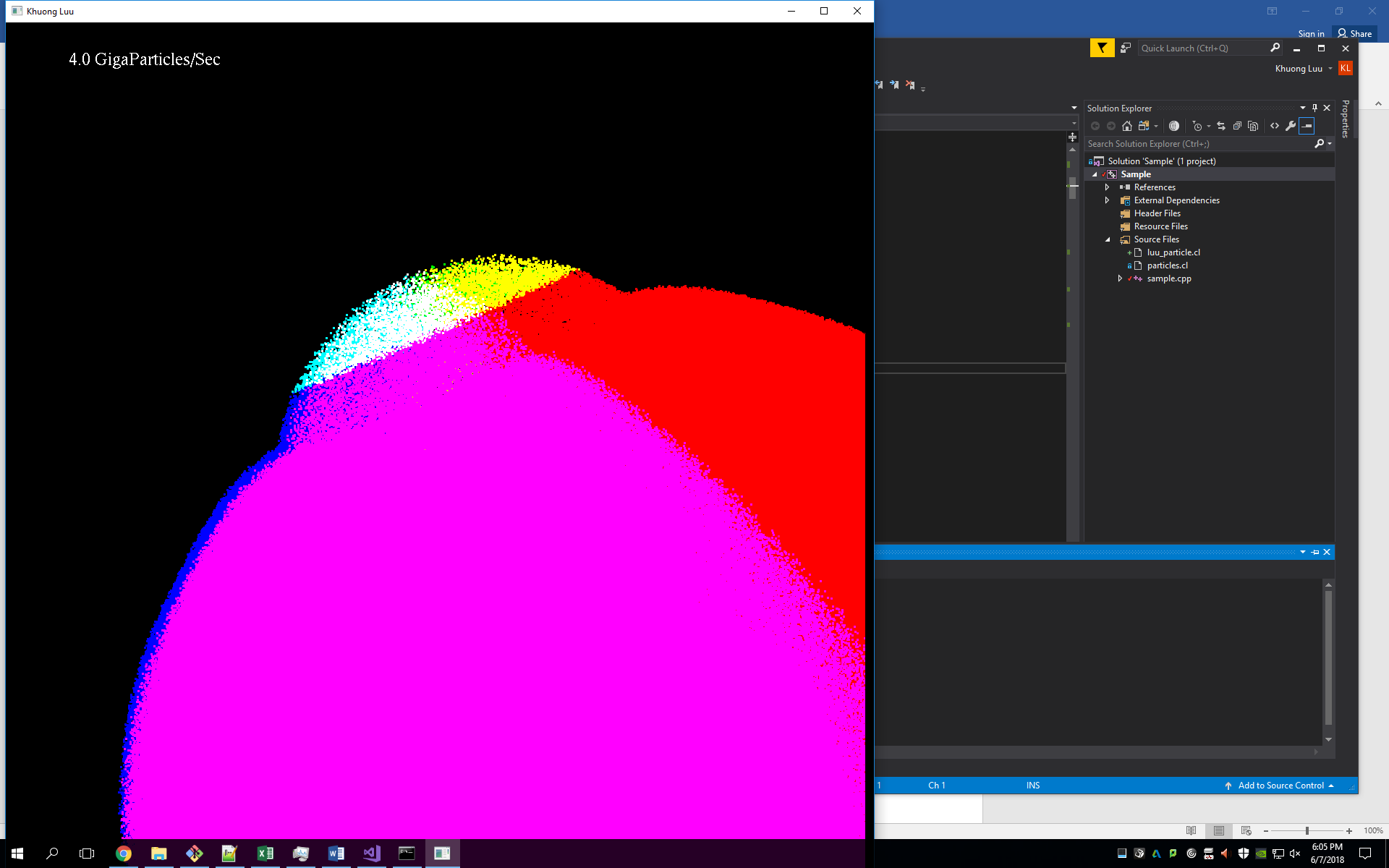
Then the particle bounce off the white sphere, its color is changed briefly to **red**

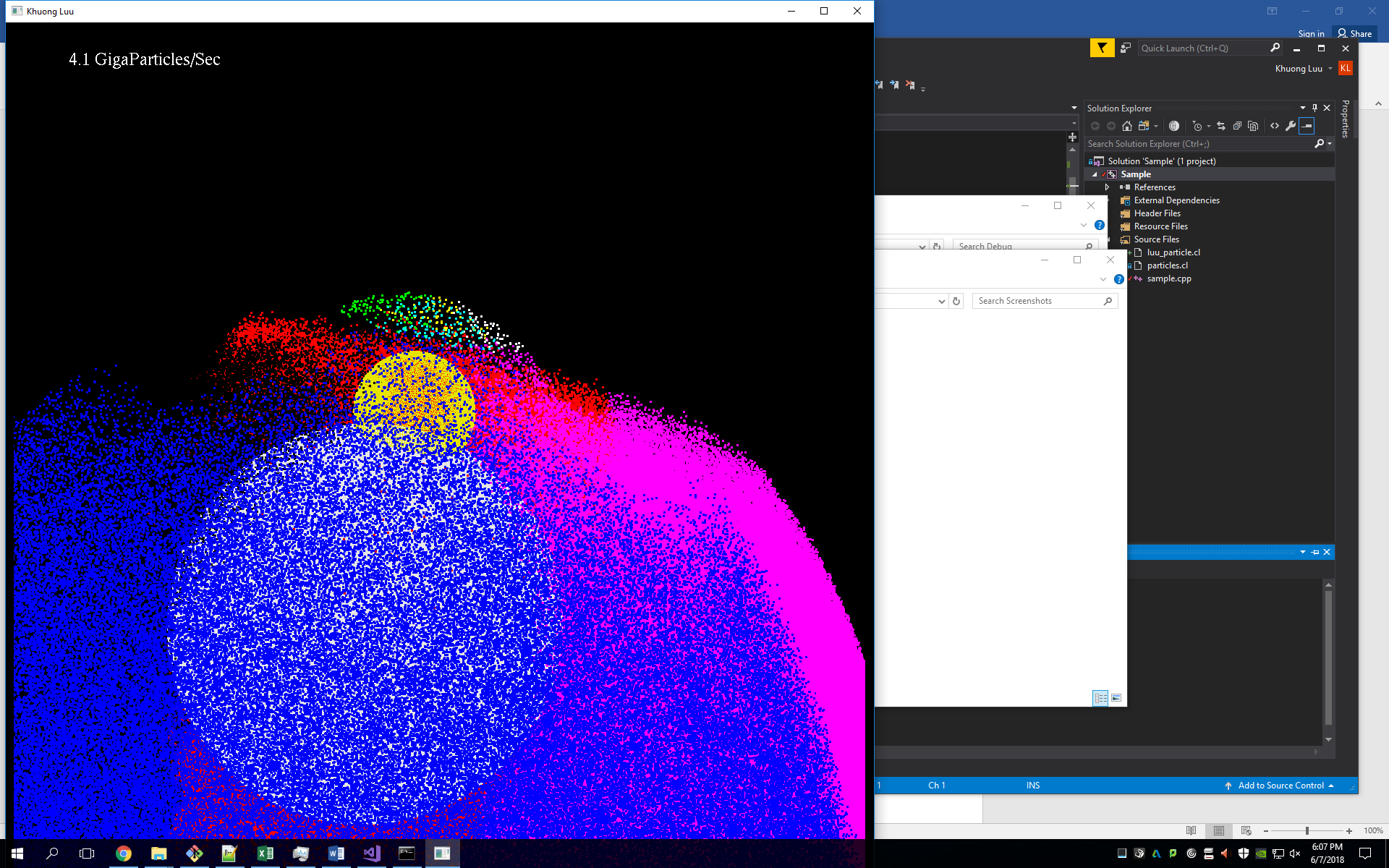
|  |
| --- |
| if (IsInsideSphere(pp, Sphere2)) {  vp = BounceSphere(p, v, Sphere2);  pp = p + vp \* DT + (float4)(.5\*DT\*DT)\*G;  c.x = 0.9f;  c.y = 0.0f;  c.z = 0.0f;  c.w = 1.;  } |

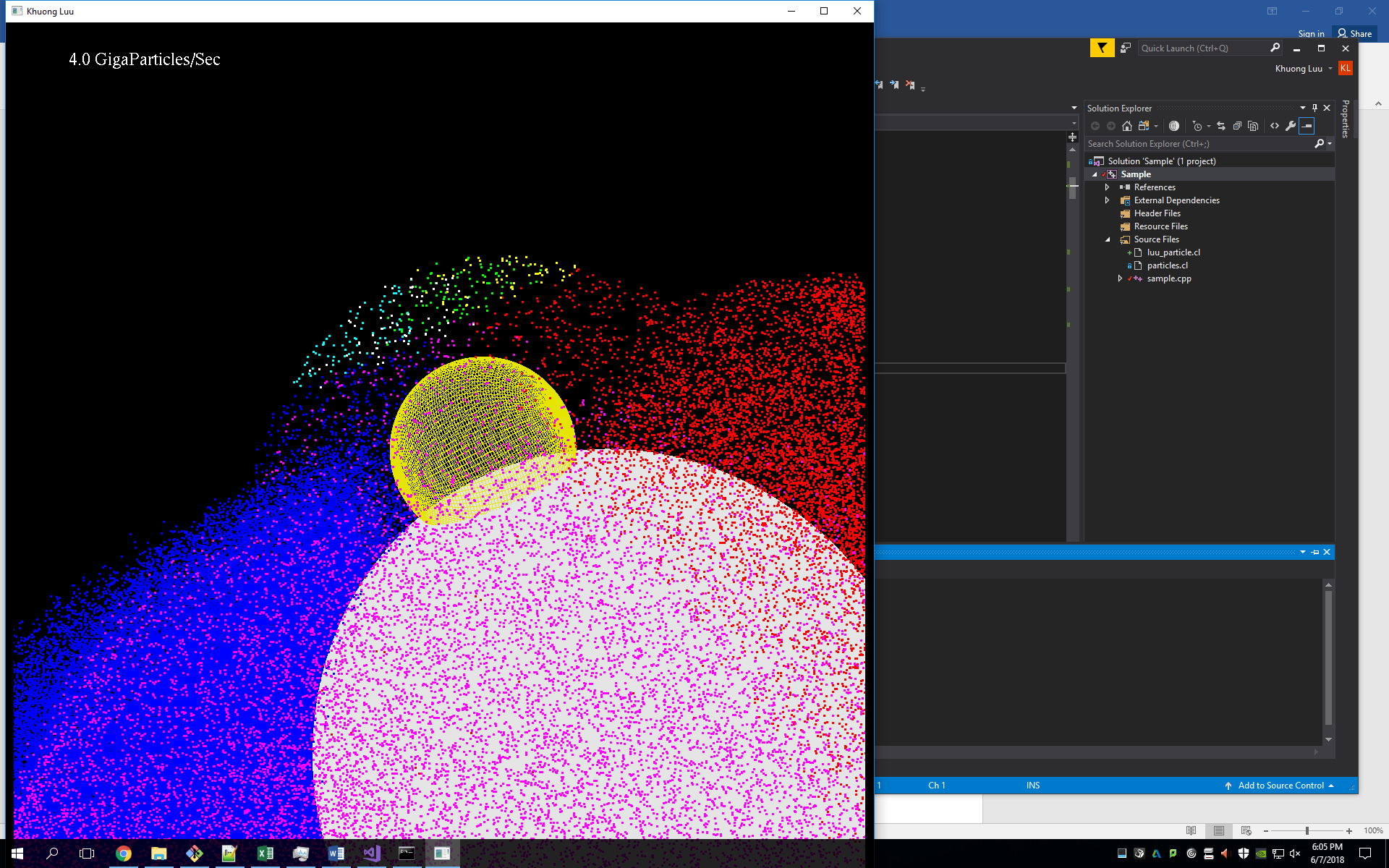
My kernel code is below:

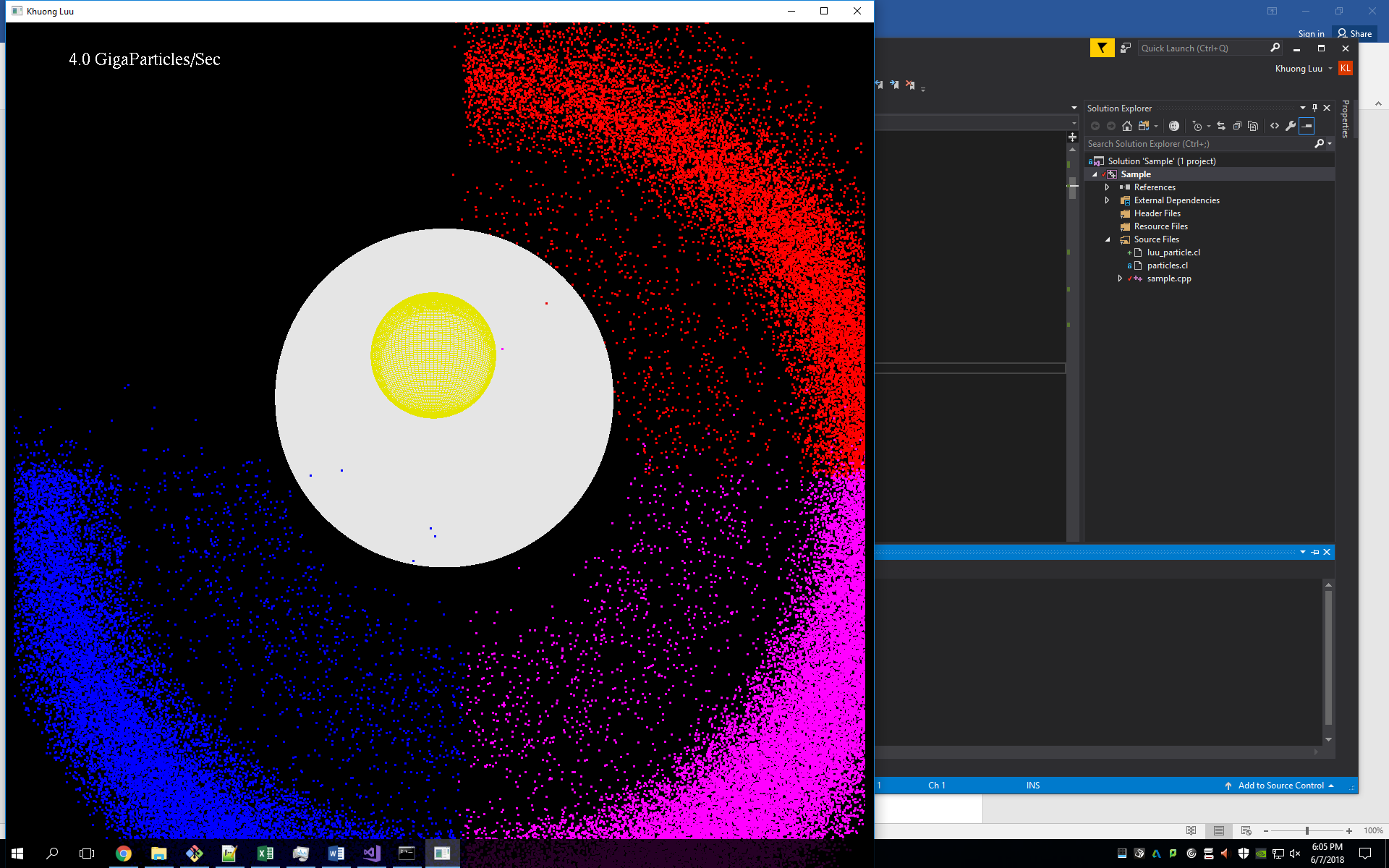
|  |
| --- |
| 1. Kernel 2. void 3. Particle( global point \*dPobj, global vector \*dVel, global color \*dCobj ) 4. { 5. const float4 G = (float4) ( 0., -9.8, 0., 0. ); 6. const float DT = 0.1; 7. const sphere Sphere1 = (sphere)( -100., -800., 0., 600. ); 8. const sphere Sphere2 = (sphere)(-100., -3000., 0., 2000); 9. int gid = get\_global\_id( 0 ); 10. point p = dPobj[gid]; 11. vector v = dVel[gid]; 12. color c = dCobj[gid]; 13. point pp = p + v\*DT + (float4)(.5\*DT\*DT)\*G; 14. vector vp = v + G\*DT; 15. c.x = vp.x + pp.x; 16. c.y = vp.y + pp.y; 17. c.z = vp.z + pp.z; 18. c.w = 1.; 19. pp.w = 1.; 20. vp.w = 0.; 21. if( IsInsideSphere( pp, Sphere1 ) ) { 22. vp = BounceSphere( p, v, Sphere1 ); 23. pp = p + vp\*DT + (float4)(.5\*DT\*DT)\*G; 24. c.x = 0.0f; 25. c.y = 0.0f; 26. c.z = 0.9f; 27. c.w = 1.; 28. } 29. if (IsInsideSphere(pp, Sphere2)) { 30. vp = BounceSphere(p, v, Sphere2); 31. pp = p + vp \* DT + (float4)(.5\*DT\*DT)\*G; 32. c.x = 0.9f; 33. c.y = 0.0f; 34. c.z = 0.0f; 35. c.w = 1.; 36. } 37. dPobj[gid] = pp; 38. dVel[gid] = vp; 39. dCobj[gid] = c; 40. } |

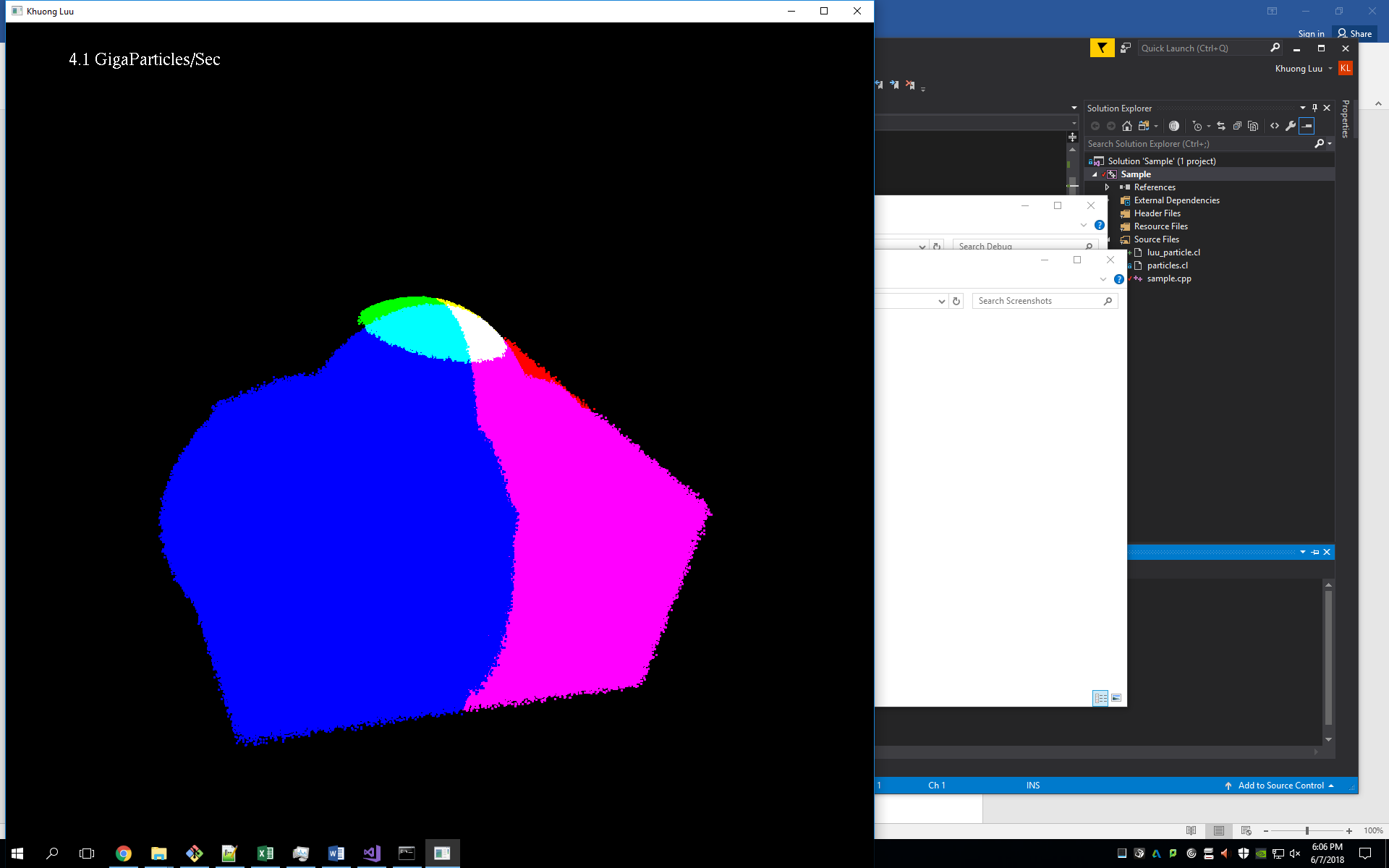
1. **Include at least one screen capture image of your project in action**

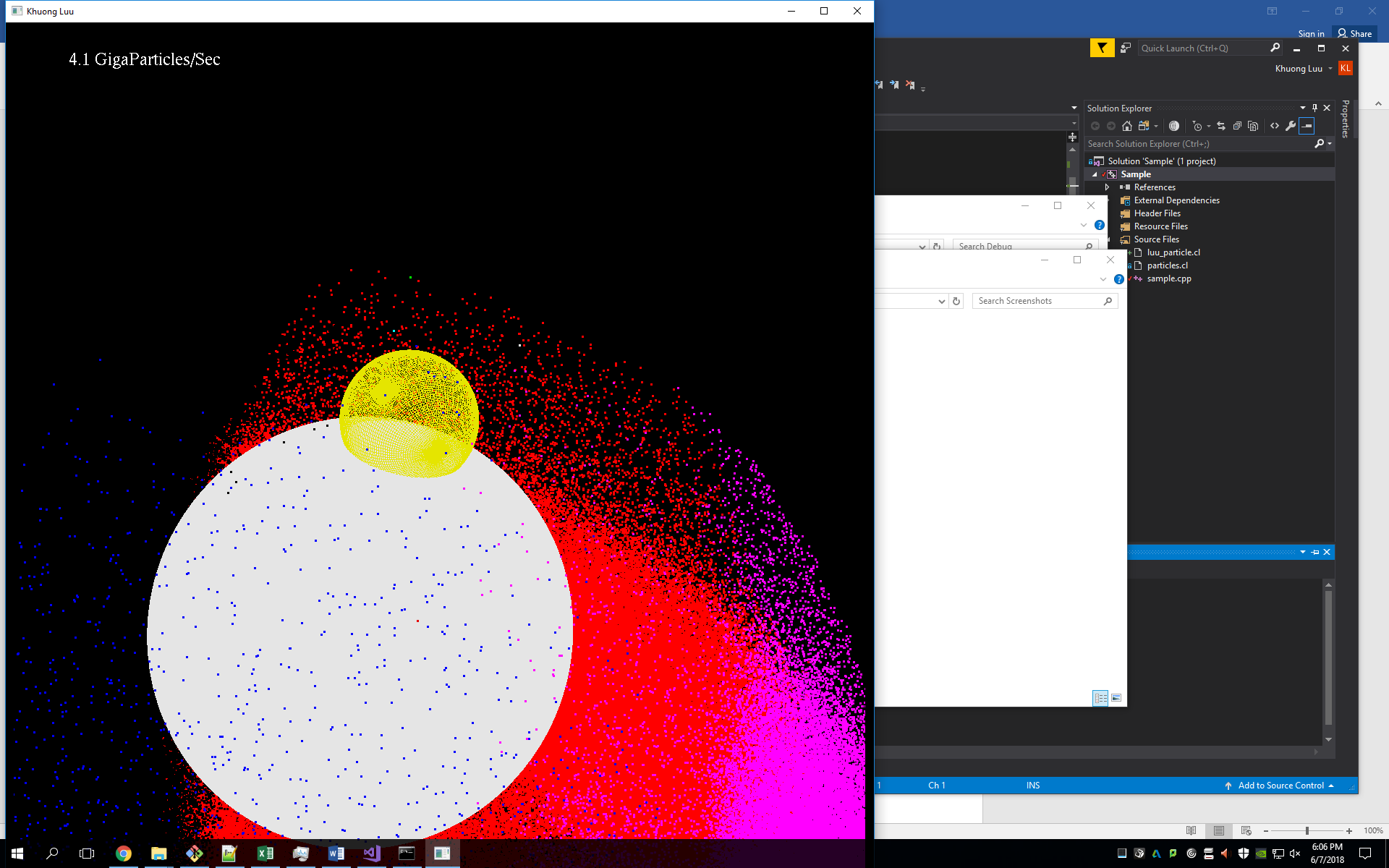
****

****

****

****

****



1. **Table**

|  |  |
| --- | --- |
| Number of Particles | Performance |
| 1024 | 0.005742 |
| 10240 | 0.060296 |
| 51200 | 0.267325 |
| 102400 | 0.468611 |
| 512000 | 1.652104 |
| 1024000 | 2.400428 |
| 5120000 | 3.815352 |
| 9216000 | 4.079692 |
| 10240000 | 4.115246 |
| 30720000 | 4.494482 |

1. Graph